

Recursie

Fie s un șir cifre binare (0 și 1) de lungime n . Definim \underline{s} ca șirul binar de lungime n , astfel încât $s[i] \neq \underline{s}[i]$, pentru fiecare i de la 1 la n . De exemplu, dacă $s = 010110$, atunci $\underline{s} = 101001$. Iar dacă $s = 0000$, atunci $\underline{s} = 1111$.

Fie a și b două șiruri de cifre binare. Definim ab concatenarea celor două șiruri. De exemplu, dacă $a = 101$ iar $b = 001$, atunci $ab = 101001$.

Recent Gabriel a învățat recursia la lecțiile de informatică și a decis să își aplice cunoștințele. El fixează un cronometru și începe să scrie șiruri de cifre binare pe tablă. În minutul 0, el scrie pe o tablă un șir de cifre binare – s_0 . Apoi, el definește următorul proces recursiv: În minutul i , $i \geq 1$, el ia șirul scris pe tablă la moment (fie s_{i-1} șirul scris pe tablă la moment), și îl înlocuiește cu șirul $s_i = s_{i-1}\underline{s_{i-1}}$. Astfel, dacă pe tablă era scris șirul $s_{i-1} = 100101$, el va fi înlocuit cu șirul $s_i = s_{i-1}\underline{s_{i-1}} = 100101011010$.

După ce trec câteva minute, Gabriel observă că aceste șiruri devin foarte lungi, și se dă bătut. Ziua următoare la lecția de informatică, el învață că un palindrom e un șir de caractere care citit de la stânga la dreapta sau de la dreapta la stânga rămâne neschimbat. De exemplu 101 , 0110 și 0 sunt palindromuri, iar 110 , 10 sau 0001 nu sunt. Acum Gabriel se întreabă, care din șirurile s_i sunt palindromuri. Gabriel își pune q întrebări. Pentru întrebarea i , el alege un indice p_i , și vrea să afle dacă s_{p_i} este un palindrom.

Sarcina. Elaborați un program care îl ajută pe Gabriel să răspundă la cele q întrebări.

Date de intrare: Prima linie a intrării standard conține un șir de cifre binare (0 și 1) - s_0 . A doua linie conține un număr întreg q . Fiecare din următoarele q linii conține câte un număr întreg p_i .

Date de ieșire: Să se afișeze q linii, a i -a linie conține un singur cuvânt – “Da” sau “Nu”, dacă s_{p_i} este un palindrom sau nu.

Restricții: $|s_0| \leq 10^5$, $1 \leq q \leq 10$, $0 \leq p_i \leq 10^9$. Restricțiile referitoare la timpul de execuție și volumul utilizat de memorie sunt date în descrierea generală a problemelor propuse pentru rezolvare. Fișierul sursă va avea denumirea `recursie.pas`, `recursie.c` sau `recursie.cpp`.

Punctare: Testele sunt împărțite în mai multe grupuri (subtaskuri), fiecare grup având sub-restricții particulare.

Grup	Procentaj teste	Restricții
1 (Subtask 1)	0 (Exemple)	Restricții de bază
2 (Subtask 2)	20%	$ s_0 \leq 2$
3 (Subtask 3)	20%	$p_i \leq 3$
4 (Subtask 4)	60%	Fără restricții adiționale

Exemplu 1.

Intrare

```
01
4
0
1
2
1000000000
```

Ieșire

```
Nu
Da
Nu
Nu
```

Exemplu 2:*Intrare*

```
11
4
0
1
2
1000000000
```

Ieșire

```
Da
Nu
Da
Da
```

Exemplu 3:*Intrare*

```
100
4
0
1
2
1000000000
```

Ieșire

```
Nu
Nu
Nu
Nu
```

Explicație:

În primul exemplu, șirurile vor fi:

$s_0 = 01$, nu este palindrom

$s_1 = 0110$, este palindrom

$s_2 = 01101001$, nu este palindrom